UNIVERSITÄT BERN

Faculty of Economics and Social Sciences

Department of Economics

Applying Markowitz's Critical Line Algorithm

Andras Niedermayer
Daniel Niedermayer

06-02

March 2006

DISCUSSION PAPERS

Schanzeneckstrasse 1
Postfach 8573
CH-3001 Bern, Switzerland
http://www.vwi.unibe.ch

Applying Markowitz's Critical Line Algorithm*

Andras Niedermayer† Daniel Niedermayer‡

March 7, 2006

Abstract

We provide a Matlab quadratic optimization tool based on Markowitz's citical line algorithm that significantly outperforms standard software packages and a recently developed operations research algorithm. As an illustration: For a 2000 asset universe our method needs less than a second to compute the whole frontier whereas the quickest competitor needs several hours. This paper can be considered as a didactic alternative to the critical line algorithm such as presented by Markowitz and treats all steps required by the algorithm explicitly. Finally, we present a benchmark of different optimization algorithms' performance.

Keywords: finance, portfolio selection, efficient frontier, critical line algorithm, quadratic optimization, numerical methods
JEL-Classification: C15, C61, C63, G11

*The authors thank Ferenc Niedermayer and Heinz Zimmermann for very helpful comments. We further thank G. Peter Todd for providing us the Excel implementation of the critical line algorithm from Markowitz and Todd (2000).

†Economics Department, University of Bern, Schanzeneckstrasse 1, CH-3001 Bern, Switzerland. Email: niedermayer@vwi.unibe.ch.

‡WWZ, University of Basel, Holbeinstrasse 12, CH-4051 Basel, Switzerland. Email: daniel.niedermayer@vwi.unibe.ch.

Introduction

Markowitz's (1952) Portfolio Theory formulates investors' decisions in a mean-variance setting as a problem of minimizing portfolio variance at a certain level of expected return. The solution set of this problem is visualized by the minimum variance frontier and its positively sloped segment, the *efficient frontier*.

When including the practically relevant condition that short selling cannot take place – thus, that investors cannot weight assets negatively in portfolios– the system of linear equations is extended by n (weak) inequalities, one for each asset. Minimizing portfolio variance with equality and inequality conditions requires expensive quadratic optimization algorithms such as first stated in the critical line algorithm (CLA) of Markowitz (1956) and (1959) and the extended simplex algorithm of Wolfe (1959).

Current research by Steuer, Qi, and Hirschberger (2006) indicate the still remaining need of improving quadratic optimization algorithms' performance. They develop a simplex based algorithm that calculates all turning points of the constrained minimum variance frontier while significantly reducing computational time compared to standard software packages such as Matlab, Cplex, LINGO, Mathematica and premium Solver. When comparing their algorithm's performance with the VBA based implementation of the Optimizer by Markowitz and Todd (2000) they encounter the problem of the 256 column limitation of MS Excel. In order to circumvent this problem we implemented a similar algorithm as in Markowitz and Todd (2000) in Fortran 90 and show that this algorithm outperforms the algorithm in Steuer, Qi, and Hirschberger (2006) by a factor of almost 10 thousand (for 2000 assets) and standard software packages by even more.

From this observation we conclude that the high performance of the CLA is not well known. In fact, excluding the paper by Steuer, Qi, and Hirschberger (2006), no studies benchmarking quadratic optimization algorithms' performance are known to us. Moreover, as no publicly available software package exists that computes the entire constrained minimum variance frontier, we provide a Matlab optimization package using our Fortran 90 implementation of the CLA.

Finally, this paper can be considered as a didactic alternative to the standard CLA as presented by Markowitz. All numerical improvements to the algorithm are treated explicitly.

The rest of the paper is organized as follows: The first section introduces the mathematical framework and definitions required by the CLA. The second section formulates the quadratic optimization method and the numerical improvement. Finally, performance tests are conducted and computational experience is reported.

1 The Framework

Given is a universe of n assets with

$\boldsymbol{\Sigma}$: an $(n \times n)$ positive definite covariance matrix

$\boldsymbol{\mu}$: n vector with the assets' expected returns,

$\boldsymbol{w}$: n vector with the assets' weights.

In any minimum variance portfolio where negative weights are disallowed we define

$\mathbb{S}$: A subset of $\{1, 2, \ldots, n\}$ containing all assets' indices where weights are non-zero (IN variables in the notation of Jacobs, Levy, and Markowitz (2005)),

k: number of elements in $\mathbb{S}$,

$\boldsymbol{\Sigma_S}$: a $(k \times k)$ covariance matrix of the non-zero weighted assets

$\boldsymbol{\mu_S}$: k vector with the non-zero weighted assets' expected returns,

$\boldsymbol{w_S}$: k vector of the non-zero weighted assets weights.

1.1 Unconstrained Case

Before turning to the constrained variance minimization it is worth to familiarize oneself again with the unconstrained portfolio minimization problem. The unconstrained problem can be written as a Lagrange function

$$L = \frac{1}{2}\boldsymbol{w'\Sigma w} - \gamma(\boldsymbol{w'1} - 1) - \lambda(\boldsymbol{w'\mu} - \mu_p)\,, \tag{1}$$

with the Lagrange coefficients γ and λ and the expected return level μ_p. The first constraint in (1) ensures that assets' weights sum to one; the second constraint tells that portfolio variance is minimized at the expected return level of μ_p. Differentiating with respect to $\boldsymbol{w}$, γ and λ and setting the results to zero, one obtains a system of $(n + 2)$ linear equations. Solving this system leads to the solution of the variance minimizing weight vector $\boldsymbol{w^*}$. Obviously, $\boldsymbol{w^*}$ will generally not contain only positive weights and will not satisfy the constrained minimization problem.

1.2 Constrained Case

If constraints against short selling, thus, against negative portfolio weights are imposed, the solution for $\boldsymbol{w}^*$ cannot be written in a simple form and quadratic optimization algorithms have to be used. In the following, we shall call the solution of this problem a constrained minimum variance portfolio[1] and the graphical representation of the set of solutions in the (μ_p, σ_p) plane as constrained minimum variance frontier (CMVF).

One important feature of the CMVF is the existence of turning points.[2]

Definition 1 A constrained minimum variance portfolio is called *turning point* if in its vicinity other constrained minimum variance portfolios contain a different number of non-zero weighted assets. □

When knowing which assets at a certain expected return level μ_p are non-zero in the constrained minimum variance portfolio, thus, knowing $\mathbb{S}$, the problem can be formulated easily. This is stated in the following proposition.

Proposition 1 *The non-zero weights in the solution* $\boldsymbol{w}^*$ *of the constrained case are equal to the weights in the solution of the unconstrained case with the Lagrange function*

$$L = \frac{1}{2}\boldsymbol{w}_{\boldsymbol{S}}'\boldsymbol{\Sigma}_{\boldsymbol{S}}\boldsymbol{w}_{\boldsymbol{S}} - \gamma(\boldsymbol{w}_{\boldsymbol{S}}'\mathbf{1}_{\boldsymbol{S}} - 1) - \lambda(\boldsymbol{w}_{\boldsymbol{S}}'\boldsymbol{\mu}_{\boldsymbol{S}} - \mu_p)\,, \tag{2}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{S}}$, $\boldsymbol{\mu}_{\boldsymbol{S}}$, $\mathbf{1}_{\boldsymbol{S}}$ *and* $\boldsymbol{w}_{\boldsymbol{S}}$ *are the corresponding objects restricted to the subset* $\mathbb{S}$.

[1] This is also called a feasible mean-variance efficient portfolio in the literature.

[2] In Markowitz (1959) turning points are described as the intersections of two critical lines.

PROOF This is obvious: changing $\boldsymbol{w_S}$ infinitesimally ensures that all weights remain positive. This cannot lead to a smaller L otherwise $\boldsymbol{w^*}$ would not be a solution of the constrained case. ■

Even if (2) looks similar to (1), there is a major difference; the underlying subset in (2) contains only the k non-zero assets of the constrained problem's solution. Since the subset $\mathbb{S}$ does not change between turning points, the solutions between two turning points will be the solution of an unconstrained optimization upon the subset $\mathbb{S}$.

Corollary 1 *Combining two neighboring turning points with a real weight* $\omega \in [0, 1]$ *always leads to a constrained minimum variance portfolio.*

PROOF This follows from Proposition 1 and the fact that the linear combination of two solutions (for different values of μ_p) of the unconstrained problem are a solution as well. ■

Differentiating (2) with respect to $\boldsymbol{w_S}$ yields

$$\boldsymbol{\Sigma_S w_S} - \lambda \boldsymbol{\mu_S} = \gamma \mathbf{1}\,. \tag{3}$$

Making use of the constraint $\mathbf{1'w_S} = 1$ the value of γ in (3) can be calculated as

$$\gamma = \frac{1}{\mathbf{1}'_{\boldsymbol{S}} \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1} \mathbf{1}_{\boldsymbol{S}}} - \frac{\mathbf{1}'_{\boldsymbol{S}} \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1} \boldsymbol{\mu}_{\boldsymbol{S}}}{\mathbf{1}'_{\boldsymbol{S}} \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1} \mathbf{1}_{\boldsymbol{S}}} \lambda\,. \tag{4}$$

Note that here λ is set exogenously instead of μ_p. Therefore, λ determines the value of γ and finally the expected return of the minimum variance portfolio. The value of μ_p in (2) is fictitious in this case and the optimal solution is solely determined by λ. In fact, it is very similar to set λ exogenously or to calculate with a fixed μ_p and look at λ as Lagrange multiplier. This is

because λ and $\boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{w}_{\boldsymbol{S}}$ $(= \mu_p)$ are linearly related within turning points and because a higher λ yields a (constrained) minimum variance portfolio with higher expected return. This is stated here by Proposition 2 with the proof being banned to the appendix.

Proposition 2 *Between two turning points λ and $\boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{w}_{\boldsymbol{S}}$ are linearly related with a positive slope*

$$\frac{\partial(\boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{w}_{\boldsymbol{S}}(\lambda))}{\partial\lambda} > 0\,. \qquad \square$$

2 The Algorithm

The main idea for the presented algorithm is the following: first, the turning point with the lowest expected return value is found; then the next higher turning point is calculated. This is shown in Figure 1.

From the definition of a turning point we know that each of them will differ in the composition of its non-zero weighted assets. Therefore, for each of them (3) will hold for a different subset $\mathbb{S}$. Except for the case where two or more turning points lie upon each other[3], the number of non-zero weighted assets in two neighboring turning points differ by exactly one.

Moreover, when moving upwards from a turning point to the next one, λ will increase (see Proposition 2). When looking at turning points such as in Figure 1 it must therefore be that

$$\lambda_1 < \lambda_2 < \lambda_3 < \cdots < \lambda_Q\,.$$

[3]We exclude this possibility; when calculating numerically, there will hardly be two or more turning points on one (σ, μ) location. Markowitz (1959) proposes as a solution to this problem to either alter the μ of one asset or to use the method described in Markowitz (1956).

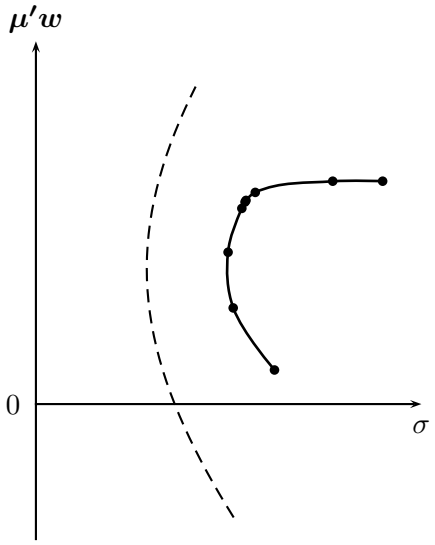

Figure 1: This figure shows the minimum variance frontier (dashed line) and the constrained minimum variance frontier for ten assets and an arbitrarily chosen non-singular covariance matrix. The dots represent the constrained frontier's turning points.

The next two sections show how to find turning point 1 and how to move to the next higher turning point. The third section shows a way how to improve the algorithm's performance significantly.

2.1 Starting Solution

This algorithm requires an initial solution on the constrained minimum variance frontier. Let j be the asset's index that has the minimal expected return, $\mu_j = \min\{\mu_1, \mu_2, \ldots, \mu_n\}$ and $\boldsymbol{w}^{(1)}$ defined as

$$w_j^{(1)} = 1 \tag{5}$$

$$w_i^{(1)} = 0 \qquad i = 1 \ldots n\,, i \neq j \tag{6}$$

The obtained weight vector $\boldsymbol{w}^{(1)}$ describes the lowest turning point such as turning point 1 in Figure 1. Moving infinitesimally upwards in $\boldsymbol{\mu' w}$ will increase the weight of one of the zero weighted assets.

2.2 Iteration

When moving from a turning point to the next higher one, one of the following two situations must occur; either one non-zero weighted asset becomes zero or a formerly zero weighted asset becomes non-zero. This two cases must be looked at in order to compute the next turning point's λ and $\boldsymbol{w_S}$.

Case a) one formerly non-zero weighted asset becomes zero.[4]

Let λ_{current} correspond to a turning point and let $\mathbb{S}$ be the set of the non-zero weighted assets slightly above this turning point (i.e. for λ such that $\lambda_{\text{current}} \equiv \lambda_l < \lambda < \lambda_{l+1}$). For this subset containing k variables equation (3) holds and thus

$$\boldsymbol{w_S} = \lambda \boldsymbol{\Sigma_S^{-1}} \boldsymbol{\mu_S} + \gamma \boldsymbol{\Sigma_S^{-1}} \mathbf{1_S} \,. \tag{7}$$

For a given subset $\mathbb{S}$ and an asset $i \in \mathbb{S}$ the value of λ (and $\gamma = \gamma(\lambda)$ according to (4)) can be calculated where the weight of asset i is zero. These asset (and subset) specific values are denoted as $\lambda^{(i)}$ and $\gamma^{(i)}$ in the following.

We will further denote the ith componenent of $\boldsymbol{w_S}$ as w_i. Setting w_i in (7) to zero yields

$$w_i = 0 = \lambda^{(i)} (\boldsymbol{\Sigma_S^{-1}} \boldsymbol{\mu})_i + \gamma^{(i)} (\boldsymbol{\Sigma_S^{-1}} \mathbf{1})_i \,. \tag{8}$$

Note that $i \in \mathbb{S} = \{i_1, i_2, \ldots, i_k\}$ and therefore can take values from 1 to n (and not from 1 to k).

[4] Or in other words: a critical line cuts another critical line with one asset less.

Solving (8) for $\lambda^{(i)}$ leads to

$$\lambda^{(i)} = \frac{(\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}})_i}{\mathbf{1}'_{\boldsymbol{S}}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}}(\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}})_i - \mathbf{1}'_{\boldsymbol{S}}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}(\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}})_i}\,. \tag{9}$$

The next $\lambda > \lambda_{\text{current}}$ where an asset wants to leave the subset $\mathbb{S}$ is[5]

$$\lambda_{\text{inside}} = \min_i\{\lambda^{(i)} \mid \lambda^{(i)} > \lambda_{\text{current}}\}\,, \qquad i \in \mathbb{S}\,. \tag{10}$$

However, λ_{inside} will only describe the next higher turning point if there is no portfolio with a λ where $\lambda_{\text{current}} < \lambda < \lambda_{\text{inside}}$ and where an asset wants to get into the subset $\mathbb{S}$. This situation is summarized by case b. Moreover, if no $\lambda^{(i)} > \lambda_{\text{current}}$ could be found we remember that no solution for λ_{inside} does exist.

Case b) one of the formerly zero weighted assets wants to become positive.[6]

When moving upwards in $\boldsymbol{\mu'w}$ it might occur that a formerly zero weighted assets i wants to become non-zero. The corresponding portfolio must therefore be a turning point where the subset $\mathbb{S}$ must be redefined by including the new asset i. Let us denote the new subset as

$$\mathbb{S}_i = \mathbb{S} \cup \{i\}.$$

Including a formerly zero weighted asset i into $\mathbb{S}$ means that $i \notin \mathbb{S}$.

Analogously to (9) the value $\lambda^{(i)}$ where w_i becomes zero[7] is given by

$$\lambda^{(i)} = \frac{(\boldsymbol{\Sigma}_{\boldsymbol{S_i}}^{-1}\mathbf{1}_{\boldsymbol{S_i}})_i}{\mathbf{1}'_{\boldsymbol{S_i}}\boldsymbol{\Sigma}_{\boldsymbol{S_i}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S_i}}(\boldsymbol{\Sigma}_{\boldsymbol{S_i}}^{-1}\mathbf{1}_{\boldsymbol{S_i}})_i - \mathbf{1}'_{\boldsymbol{S_i}}\boldsymbol{\Sigma}_{\boldsymbol{S_i}}^{-1}\mathbf{1}_{\boldsymbol{S_i}}(\boldsymbol{\Sigma}_{\boldsymbol{S_i}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S_i}})_i}. \tag{11}$$

[5]Note that if an asset i has entered the set $\mathbb{S}$ in the previous step then for this one has $\lambda^{(i)} = \lambda_{\text{current}}$. To avoid the related numerical uncertainty one can simply leave out the corresponding $\lambda^{(i)}$ from the set considered.

[6]Or: the critical line cuts another critical line with one asset more.

[7]Note that $w_i < 0$ for $\lambda : \lambda_{\text{current}} < \lambda < \lambda^{(i)}$ since $i \notin \mathbb{S}$.

In order to find the minimal $\lambda^{(i)} > \lambda_{\text{current}}$ where a currently zero-weighted asset i wants to become non-zero (11) must be applied for all $i \notin \mathbb{S}$.

$$\lambda_{\text{outside}} = \min_i \{\lambda^{(i)} \mid \lambda^{(i)} > \lambda_{\text{current}}\}\,, \qquad i \notin \mathbb{S}\,. \tag{12}$$

Again[8], if no $\lambda^{(i)} > \lambda_{\text{current}}$ exists, we remember that there is no solution for λ_{outside}.

Finding the next turning point.

In order to find out which case will occur, the values of λ_{inside} and λ_{outside} must be compared.

- If solutions for both λ_{inside} and λ_{outside} could be found, then the next turning point will have a λ defined as

  $$\lambda_{\text{new}} = \min\{\lambda_{\text{inside}}, \lambda_{\text{outside}}\}\,.$$

  Thus, e.g. case a. is characterized by $\lambda_{\text{inside}} < \lambda_{\text{outside}}$ and vice versa.

- If a solution only for λ_{inside} or λ_{outside} could be found, λ_{new} is overwritten by the respective value.

- Depending on which case occurs we replace $\mathbb{S}$ by $\mathbb{S}\backslash\{i\}$ or by $\mathbb{S}_i$ and λ_{current} by λ_{new}.

- If no solution for λ_{inside} and λ_{outside} could be found, we have reached the highest turning point and the algorithm terminates.

One way of checking the results is to look at the last turning point's weight vector. This weight vector must be the 'opposite' one to the initial

[8] Similarly to case a, if some asset i first left the set of positively weighted assets in the previous step, it will have $\lambda^{(i)} = \lambda_{\text{current}}$, and it should not be considered.

solution with $w_j = 1$, $w_i = 0$, $i = 1 \ldots n, i \neq j$ and where j comes from $\mu_j = \max\{\mu_1, \mu_2, \ldots, \mu_n\}$.

Note that in contrast to the calculations in (9) the specification of $\mathbb{S}_i$ in (11) depends on i and $\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-\mathbf{1}}$ must be recalculated for each $i \notin \mathbb{S}$.[9] Moreover, when the next turning point contains a new asset or if one asset leaves $\mathbb{S}$, the inverse of the respective covariance matrix, $\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-\mathbf{1}}$ must be recalculated. However these time consuming inversions can be avoided. This procedure is described in the following.

2.3 Improving Performance

In the algorithm described above, the composition of $\mathbb{S}$ and $\mathbb{S}_i$ change. There is always either one asset included or excluded from the subset which means that a row and a column are appended or deleted from the covariance matrix. In both cases the inverse of the respective matrices do not have to be recalculated by a matrix inversion. This is shown in the following.

Expansion of the covariance matrix.

Lemma 1 *Let $\boldsymbol{A}$ be a symmetric non-singular $k \times k$ matrix, $\boldsymbol{a}$ an $k \times 1$ vector and α a scalar. Then for the expanded matrix's inverse*

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{a'} & \alpha \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A^{-1}} + \beta \boldsymbol{c}\boldsymbol{c'} & -\beta \boldsymbol{c} \\ -\beta \boldsymbol{c'} & \beta \end{bmatrix} \tag{13}$$

holds where

$$\boldsymbol{c} = \boldsymbol{A^{-1}}\boldsymbol{a} \qquad \textit{and} \qquad \beta = \frac{1}{\alpha - \boldsymbol{c'a}}\,.$$

PROOF Multiplying the expanded matrix with the right-hand side of (13) yields the identity matrix. ■

[9] Or at least the vectors $\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-\mathbf{1}}\mathbf{1}_{\boldsymbol{S}_i}$ and $\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-\mathbf{1}}\boldsymbol{\mu}_{\boldsymbol{S}_i}$ have to be calculated.

Our algorithm requires often expanding the subset $\mathbb{S}$ by one element and recalculating the inverse of the covariance matrix for the new subset. Lemma 1 frees us from the burden of making this calculation all over again. This reduces the number of operations for inverting $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{a'} & \alpha \end{bmatrix}$ from $k^3/3$ to $2k^2$.

Reduction of the covariance matrix.

Reducing the covariance matrix by one row and column does not require the inversion of the newly obtained matrix either. Having calculated the inverse of the expanded covariance matrix as in the previous section and now deleting the last[10] row and column, the newly obtained matrix's inverse can be calculated. This is stated in Lemma 2.

Lemma 2 *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be $k \times k$ matrices, $\boldsymbol{a}$ and $\boldsymbol{b}$ k vectors and α and β two scalars. Then if*

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{a'} & \alpha \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{b} \\ \boldsymbol{b'} & \beta \end{bmatrix} \tag{14}$$

holds then

$$\boldsymbol{A^{-1}} = \boldsymbol{B} - \frac{1}{\beta}\boldsymbol{b}\boldsymbol{b'}\,.$$

holds as well.

PROOF By combining (13) and (14) and solving for $\boldsymbol{A^{-1}}$. ∎

Key Improvement.

The most remarkable improvement stems from the fact that we do not need to know the inverse of $\boldsymbol{\Sigma_{S_i}}$ in (11) in order to find the minimal $\lambda^{(i)}$. This is stated in Proposition 3.

[10]This can be made by redefinition of the variables without loss of generality.

Proposition 3 *Expression* (11)*,*

$$\lambda^{(i)} = \frac{(\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\mathbf{1}_{\boldsymbol{S}_i})_i}{\mathbf{1}'_{\boldsymbol{S}_i}\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}_i}(\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\mathbf{1}_{\boldsymbol{S}_i})_i - \mathbf{1}'_{\boldsymbol{S}_i}\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\mathbf{1}_{\boldsymbol{S}_i}(\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}_i})_i} \tag{15}$$

can be rewritten as

$$\lambda^{(i)} = \frac{1 - \boldsymbol{a}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}}{\boldsymbol{\mu}'_{\boldsymbol{S}}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}(1 - \boldsymbol{a}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}) - \mathbf{1}'_{\boldsymbol{S}}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}(\mu_i - \boldsymbol{a}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}})} \tag{16}$$

where the expanded covariance matrix and the expanded return vector are

$$\boldsymbol{\Sigma}_{\boldsymbol{S}_i} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{S}} & \boldsymbol{a} \\ \boldsymbol{a}' & \alpha \end{bmatrix}, \tag{17}$$

$$\boldsymbol{\mu}_{\boldsymbol{S}_i} = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{S}} \\ \mu_i \end{bmatrix}. \tag{18}$$

Proof According to Lemma 1 $\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}$ can be expressed in terms of $\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}$, $\boldsymbol{a}$ and α. Multiplying $\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}$ by $\mathbf{1}_{\boldsymbol{S}_i}$ and $\boldsymbol{\mu}_{\boldsymbol{S}_i}$ respectively yields

$$\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\mathbf{1}_{\boldsymbol{S}_i} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}} - \beta(1 - \boldsymbol{c}'\mathbf{1})\boldsymbol{c} \\ \beta(1 - \boldsymbol{c}'\mathbf{1}) \end{bmatrix} \tag{19}$$

and

$$\boldsymbol{\Sigma}_{\boldsymbol{S}_i}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}_i} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}} - \beta(\mu_i - \boldsymbol{c}'\boldsymbol{\mu}_{\boldsymbol{S}})\boldsymbol{c} \\ \beta(\mu_i - \boldsymbol{c}'\boldsymbol{\mu}_{\boldsymbol{S}}) \end{bmatrix}. \tag{20}$$

Multiplying (19) and (20) by $\mathbf{1}'_{\boldsymbol{S}_i}$ and plugging the values into (15) yields (16). ■

The results of Proposition 3 allow us to strongly reduce the computational costs to $2nk + 2k^2$ operations. Note that in (16) one has to calculate $n - k$ times the scalar product of $\boldsymbol{a}$ (corresponding to an $i \notin \mathbb{S}$) with the vectors $\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}$, $\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}}$ which do not change.

3 Performance Tests

Obviously, it is problematic to compare different algorithms based on absolute CPU times. Their performance will strongly depend on the programming language and on the algorithm's memory requirements.

When not testing all algorithms on the same computer, different processor performance, RAM size and system configurations do not allow for comparing CPU times directly.

However, there are two reasons for us to make such performance comparisons. First, when looking at the increase of CPU time at an increasing number of assets, the algorithms' relative performance is independent from the programming language and other hardware and software properties (as long as there are no memory bottlenecks). Second, the difference in the programming languages does not explain that amount of CPU time improvement such as obtained by our tests.

In the following a Fortran 90 implementation (Fortran 90 CLA) of the discussed algorithm is tested against three programs; a simplex like algorithm based on Wolfe (1959) coded in Java (Java Wolfe-Simplex as described and implemented in Niedermayer (2005)) and the quadratic optimization package of Matlab. Furthermore, we compare our results with those in Steuer, Qi, and Hirschberger (2006)[11], whose simplex based multiparametric optimization algorithm was implemented in Java (Java MPQ). The latter comparison is important; as argued in Steuer, Qi, and Hirschberger (2006), the MPQ outperforms Matlab, Cplex, LINGO, Mathematica, and Excel's premium Solver. Steuer, Qi, and Hirschberger (2006) did not com-

[11] Steuer, Qi, and Hirschberger (2006) run their tests on a Dell 3.06 GHz desktop as well.

pare the Java MPQ algorithm to the Excel Optimizer by Markowitz and Todd (2000) due to the 256 column limitation of Excel. Finally, we also provide run times of the Excel Optimizer by Markowitz and Todd (2000). Note that this implementation is provided by Markowitz and Todd (2000) for illustrative purposes in form of an Excel VBA macro and can calculate the efficient frontier for up to 256 securities (the maximal number of columns in Excel). Note further that even though we ran the Optimizer with the same set of constraints as the other problems, it can solve the optimization problem for a more general set of constraints.

For the tests illustrated in Figure 2 a positive definite covariance matrix was generated as

$$\boldsymbol{\Sigma} = \sum_{i=1}^{n} \boldsymbol{r}^{(i)} \boldsymbol{r}^{(i)\prime},$$

where $\boldsymbol{r}^{(i)}$ is an n vector containing random numbers between $[0, 1]$ and is regenerated for each i. Since our results and the MPQ results in Steuer, Qi, and Hirschberger (2006) strongly depend on the number of IN assets, thus, of the maximum dimension, $\hat{k}$, of $\boldsymbol{\Sigma}_{\boldsymbol{S}}$ in (9) and (11), we made sure that $\hat{k}/n$ is similar to that in Steuer, Qi, and Hirschberger (2006). In our tests with 1000 assets $\hat{k}$ was 60 and when testing with 2000 assets $\hat{k}$ was 250.

In Figure 2 both axes are logarithmic. The slope of the linear fit (of an OLS fit) corresponds therefore to the exponent of the respective algorithm's CPU time increase at an increasing number of assets. Note that the problem with Java Wolfe-Simplex is that the program's RAM requirements increase rapidly which allows only for the computation of problems up to 150 assets. The test's results are summarized in Table 1.

Since the Wolfe-Simplex algorithm and the Matlab quadratic optimization package only calculate one single point on the constrained minimum

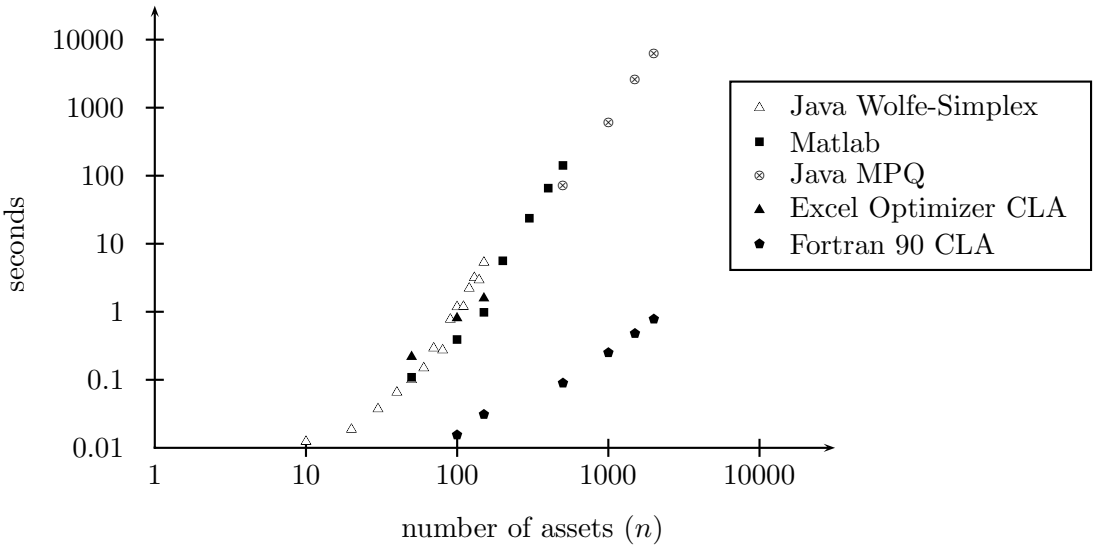

Figure 2: Testing different algorithms: CPU times for different number of assets and randomly generated positive definite covariance matrix.

n	Fortran 90 CLA	Java Wolfe-Simplex	Matlab	Java MPQ	Excel Optimizer CLA
50	-	0.10	0.109	-	0.219
100	0.0156	1.18	0.391	-	0.813
150	0.0312	5.35	0.985	-	1.578
500	0.09	-	141.6	72	-
1000	0.25	-	-	602	-
1500	0.48	-	-	2580	-
2000	0.78	-	-	6300	-
perf.	$\mathcal{O}(n^{1.55})$	$\mathcal{O}(n^{3.62})$	$\mathcal{O}(n^{3.19})$	$\mathcal{O}(n^{3.25})$	$\mathcal{O}(n^{1.81})$

Table 1: Different CPU times in seconds. The last row shows the estimates of the algorithms' performance with respect to the number of securities n. Note that the results of the MPQ performance stem from Hirschberger, Qi, and Steuer (2004). Note further that the performance we have provided for the Fortran 90 CLA is calculated from the run times without matrix sizes 100 and 150 because for smaller matrix sizes the fixed costs of calculation seem to distort the data. When including matrix sizes 100 and 150 we get $\mathcal{O}(n^{1.25})$.

variance frontier and do not calculate the whole frontier analytically such as our Fortran CLA algorithm, the Optimizer by Markowitz and Todd (2000) and the algorithm of Steuer, Qi, and Hirschberger (2006), the CPU times reported in Table 1 support our method even more.

As in Steuer, Qi, and Hirschberger (2006), our tests were conducted on a Dell desktop with a 3.06 GHz processor.

4 Conclusion

This paper presents the critical line algorithm (CLA) developed by Markowitz and demonstrates its strong computational performance compared to standard software packages and to a recently published optimization algorithm. We find that our implementation of the CLA – available on request from the authors in form of a Matlab package – outperforms the current Matlab optimization tool by a factor of approximately 15 thousand when the problem size (number of assets) is 2000. When comparing with the algorithm in Steuer, Qi, and Hirschberger (2006) that also computes all turning points analytically such as the CLA does, the performance improvement is still around 8 thousand.

As all steps of the algorithm are treated explicitly, this code can be directly used for the implementation in other programming languages and used for problems of large scale portfolio optimization.

Appendix

PROOF (PROPOSITION 2) For tractability we define three constants

$$C_{11} \equiv \mathbf{1}_{\boldsymbol{S}}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\mathbf{1}_{\boldsymbol{S}}\,, \quad C_{1\mu} \equiv \mathbf{1}_{\boldsymbol{S}}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}}\,, \quad C_{\mu\mu} \equiv \boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}}$$

Equations (3) and (4) can be written as

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{w}_{\boldsymbol{S}} &= \gamma C_{1\mu} + \lambda C_{\mu\mu} \\ \frac{\partial(\boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{w}_{\boldsymbol{S}})}{\partial\lambda} &= C_{\mu\mu} - \frac{C_{1\mu}^2}{C_{11}} \end{aligned} \tag{21}$$

Since between two turning points $\boldsymbol{\Sigma}_{\boldsymbol{S}}$ does not change, $\mu_p(\lambda) = \boldsymbol{\mu}_{\boldsymbol{S}}'\boldsymbol{w}_{\boldsymbol{S}}(\lambda)$ is linear in λ with a slope given by (21). We show below that this slope is positive.

From the positive definiteness of $\boldsymbol{\Sigma}$ follows that its submatrix $\boldsymbol{\Sigma}_{\boldsymbol{S}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}$ ($\equiv (\boldsymbol{\Sigma}_{\boldsymbol{S}})^{-1}$) are positive definite.

We introduce a vector $\boldsymbol{x} \equiv \mathbf{1}_{\boldsymbol{S}} - \alpha\boldsymbol{\mu}_{\boldsymbol{S}}$, with $\alpha \in \mathbb{R}$. Then $\boldsymbol{x}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{x}$ can be written as

$$(\mathbf{1}_{\boldsymbol{S}} - \alpha\boldsymbol{\mu}_{\boldsymbol{S}})'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}(\mathbf{1}_{\boldsymbol{S}} - \alpha\boldsymbol{\mu}_{\boldsymbol{S}}) = C_{11} - 2\alpha C_{1\mu} + \alpha^2 C_{\mu\mu}\,.$$

Positive definiteness of $\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}$ means $\boldsymbol{x}'\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{x} > 0$ for any vector $\boldsymbol{x}$, hence, the equation $C_{11} - 2\alpha C_{1\mu} + \alpha^2 C_{\mu\mu} = 0$ cannot have a solution for α. Therefore, the discriminant is negative which gives

$$C_{11}C_{\mu\mu} - C_{1\mu}^2 > 0\,.$$

■

References

HIRSCHBERGER, M., Y. QI, AND R. E. STEUER (2004): "Quadratic Parametric Programming for Portfolio Selection with Random Problem Gen-

eration and Computational Experience," Working papers, Terry College of Business, University of Georgia.

Jacobs, B. I., K. N. Levy, and H. M. Markowitz (2005): "Portfolio Optimization with Factors, Scenarios, and Realistic Short Positions," *Operations Research*, 53(4), 586–599.

Markowitz, H. M. (1952): "Portfolio Selection," *Journal of Finance*, 7(1), 77–91.

——— (1956): "The Optimization of a Quadratic Function Subject to Linear Constraints," *Naval Research Logistics Quarterly*, III, 111–133.

——— (1959): *Portfolio Selection: Efficient Diversification of Investments*. John Wiley and Sons, New York, and 1991 2nd ed., Basil Blackwell, Cambridge, MA.

Markowitz, H. M., and P. Todd (2000): *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Frank J. Fabozzi Associates, New Hope, Pennsylvania.

Niedermayer, D. (2005): "Portfolio Theory and the Cross-sectional Relation between Expected Returns and Betas," Master's thesis, University of Bern, Department of Economics.

Steuer, R. E., Y. Qi, and M. Hirschberger (2006): "Portfolio Optimization: New Capabilities and Future Methods," *Zeitschrift für BWL*, 2.

Wolfe, P. (1959): "The Simplex Method for Quadratic Programming," *Econometrica*, 27(3), 382–398.